

デコーダダウンロードシステムにおけるフレキシブルデコーダ構成

A study on Flexible Decoder Structure for Decoder Downloadable System

清水 直人*1 宮澤 敏記*1 亀山 涉*2 渡辺 裕*2 富永 英義*1*2
Naoto SHIMIZU*1 Toshinori MIYAZAWA*1 Wataru KAMEYAMA*2 Hiroshi WATANABE*2 Hideyoshi TOMINAGA*1*2

†早稲田大学 理工学部 電子・情報通信学科
†Dept. of Elec. Info. and Comm. Eng., WASEDA Univ.

‡早稲田大学大学院 国際情報通信研究科
‡Graduate School of Globa Info. and Tele. Study, WASEDA Univ.

1 はじめに

現在流通している多くの動画コンテンツは DCT ベースの動画画像符号化方式である。しかし、あらゆるコンテンツが DCT による動画画像符号化に適しているわけではなく、今後各コンテンツに最適化された符号化方式の出現が予想される [1]。このような状況の中で視聴者が、符号化方式を意識することがないシステムの開発が必要不可欠である。そのようなシステムの実現例として我々はデコーダダウンロードシステムを開発した [2]。本稿では、デコーダダウンロードシステムを踏まえ、多種多様な符号化に柔軟に対応するためのデコーダ構成法、特にシンタックス解析と復号処理の分離法について述べる。

2 デコーダダウンロードシステム

2.1 システムにおける問題点

我々が開発中のデコーダダウンロードシステムにより、デコーダを動的に配送することが可能となった。しかし、一般的なデコーダは、それを構成するモジュールが密に構成されている。したがってシンタックス・復号処理アルゴリズム変更への柔軟な対応やデコーダを構成するモジュール単位の再利用が困難であり、より柔軟でスケラビリティの高いデコーダ構成法が要求される。

3 シンタックス解析と復号処理の分離

3.1 デコーダ構成

デコーダを構成する各モジュールの再利用性を高めるためには、デコーダの処理を、それを構成する各モジュールの独立性が高まるように分離する必要がある。それぞれの独立性が高まることでデコーダが疎に構成されるので、各モジュールの交換などが容易に行える。デコーダ構成の分離については Syntax Parsing, Data Restorer, Image Reconstruction の 3 つの処理に分離する方法が提案されている [3]。この方法ではシンタックスが変更された場合は Syntax Parsing, 復号アルゴリズムが変更された場合は Data Restorer を更新すれば良く、柔軟なデコーダが開発できる。ただし、ここで示されているのはあくまで柔軟なデコーダ構成の指針であり、デコーダを開発段階からこのような仕様に沿って設計する必要がある。図 1 に MPEG1・2 Video を対象としたダウンロードシステム用の分離処理の例を示す。

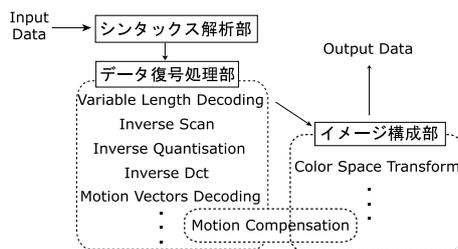


図 1 各処理の分離

このように処理を分離する場合、モジュールのインターフェースを一意に定める必要がある。図 1 におけるシンタックス解析部とデータ復号処理部間のインターフェースを一意に定める方法として MSDL-S (Mpeg4 System and Description Language - Syntactic description language) [4] を利用する。

3.2 従来型 MSDL-S コンパイラ

MSDL-S のコンパイラの例としてとして、Flavor [5] があるが、現在一般に開発されているこれらのプログラムはシンタックス解析部に対応したソースコードだけが生成される。つまり、MSDL-S を用いてデコーダを設計する場合、独自に生成されたコードに対して、データ復号処理を追加する必要がある。したがって、シンタックス解析部とデータ復号処理部とのインターフェースが独自に定義されてしま

い、シンタックス解析部を他のデコーダから共有化することが不可能である。

3.3 提案手法

データ復号処理部とイメージ構成部のインターフェースを一意に定めるには、コンパイル時にシンタックス解析部とデータ復号処理部間のインターフェースを生成すればよい。Flavor を基に提案手法の実装を行った。

MSDL-S では、関連した情報の集まり、すなわち層を class として定義する。Flavor では class 毎に対応した C++/Java におけるシンタックス解析モジュールを生成する。理想的には生成された各層に対してそれぞれの処理を記述できることが望ましい。例えば MPEG2 シンタックスの場合、ブロック層のみのデータを用いてデータ復号可能であるならば、シンタックス処理とデータ復号処理の分離は容易である。しかし、実際マクロブロック層における quantiser scale code はブロック層での DCT 係数復号に影響する。この様に各層間で独立性が保たれていない。そこで下位層での復号には上位層のデータをあらかじめ処理できるようにインターフェース用コードを生成する必要がある。しかし、MSDL-S だけではどのデータが必要であるか認識できないため上記手法は困難である。そこで、階層間で必要なデータを別ファイルとして定義し、MSDL-S と共にコンパイルすることでそのデータを下位層に渡すためのコードが自動生成される。これによりあたかも各層に対してそれぞれの処理を記述できるコードが生成できる。

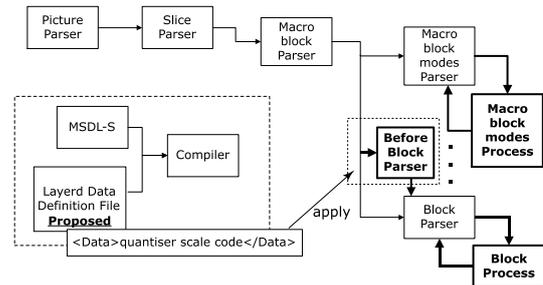


図 2 生成コード

MSDL-S と本手法を用いることでシンタックス解析部と復号処理部間のインターフェースを一意に定めることが可能であり、これにより分離が可能となる。

4 まとめ

デコーダ開発者が MSDL-S をシンタックス記述言語として採用すると共に本手法を用いることで、そのコードから自動的にシンタックス解析・復号処理を分離し、インターフェースを一意に定めることが可能となった。したがって、ダウンロードシステムにおいてはモジュールの再利用率が高まり、より効率的なモジュール伝送を実現できる。今後は本提案手法により開発したデコーダモジュールを用いて、デコーダダウンロードシステムの評価を行う予定である。

参考文献

- [1] 宮澤, 亀山, 渡辺, 阪谷, 富永 "アニメーション画像符号化の基礎検討", 通信学会全国大会, 2001
- [2] 宮澤, 清水, 亀山, 渡辺, 富永, "Pull 形サービスにおけるデコーダダウンロードシステムの構成及び評価", PCSJ, 2001
- [3] H.Arakawa, T.Maeda, M.Etoh, "Software architecture for flexible and extensible image decoding", Signal Processing: Image Communication 10, No.1-3, July 1997, pp.235-248
- [4] ISO/IEC JTC1/SC29/WG11, "ISO/IEC 14496-1"
- [5] A.Eleftheriadis, "Flavor: A Language for Media Representation", Proceedings, ACM Multimedia 97 Conference, Seattle, WA, November 1997